

#12

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-91413

(43) 公開日 平成10年(1998) 4月10日

(51) Int.Cl.<sup>8</sup>

G 0 6 F 9/06  
3/14

識別記号

5 3 0  
3 1 0

F I

G 0 6 F 9/06  
3/14

5 3 0 N  
3 1 0 E

審査請求 未請求 請求項の数13 O L (全 17 頁)

(21) 出願番号 特願平9-177305

(22) 出願日 平成9年(1997) 7月2日

(31) 優先権主張番号 9 6 1 3 7 6 7 . 4

(32) 優先日 1996年7月6日

(33) 優先権主張国 イギリス (G B)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)

(72) 発明者 ヴァネッサ・ドネリー

イギリス国 CV3 5RN ウォリック  
シャー州コヴェントリー ステイヴシェール  
ダインツリー クロフト 14

(74) 代理人 弁理士 坂口 博 (外1名)

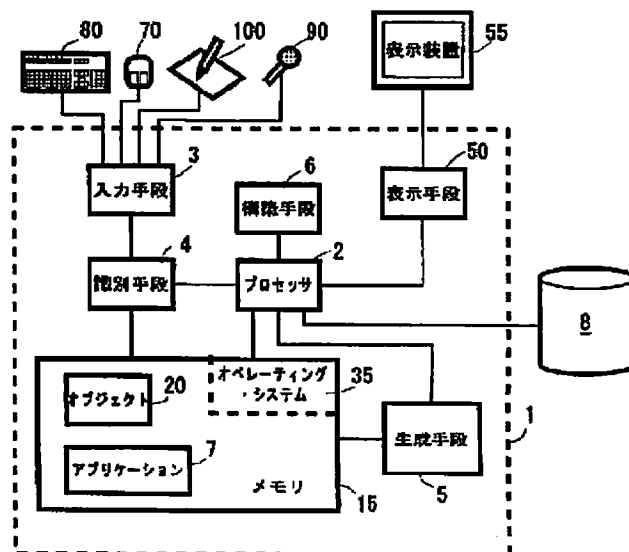
最終頁に続く

(54) 【発明の名称】 コンピュータ・システムにおける操作の選択装置及び方法

(57) 【要約】

【課題】 ユーザ・インタフェースを有し、所望の操作を呼び出すために該ユーザ・インタフェースを介してユーザが動作を選択することが可能なコンピュータ・システム、および該コンピュータ・システムを操作する方法を提供する。

【解決手段】 動作オブジェクトと動作の選択によって呼び出される操作を識別する識別子とを提供する手段と、を備え、さらに識別子を使用して所望の操作を呼び出すコマンドを生成するために動作オブジェクトは識別手段にตอบสนองすることを特徴とするコンピュータ・システム。動作は、ユーザ・インタフェースを介して1つまたはそれ以上のビューから選択可能なものとする。また、システムは動作を表現するために該動作を表現するのに用いることが可能な一以上のビューの各々について該ビューを与えるのに必要とされる属性を定義する動作オブジェクトによって動作を表現するための生成手段と、動作の選択によって呼び出される操作を識別する識別子とを有する。



## 【特許請求の範囲】

【請求項1】 ユーザ・インタフェースを有し、かつ該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、さらに前記動作は前記ユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システムであって、

動作を表現するために前記動作を表現するのに用いることが可能な前記一以上のビューの各々について該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと前記動作の選択によって呼び出される操作を識別する識別子とを提供する手段と、

ユーザによるビューの選択がいつ生じたかを識別し、前記選択に対応する動作オブジェクトを伝える識別手段とを備え、さらに前記識別子を使用して前記所望の操作を呼び出すコマンドを生成するために前記動作オブジェクトは前記識別手段に回答することを特徴とするコンピュータ・システム。

【請求項2】 前記コンピュータ・システムはグラフィカル・ユーザ・インタフェースを表示する表示装置に接続可能であり、また前記動作を表現するのに使用される前記ビューの少なくとも一つは、前記グラフィカル・ユーザ・インタフェースを介して表示可能なビジュアル・ビューであり、さらに前記ビューの少なくとも一つに対する前記動作オブジェクトに定義された前記属性に該ビジュアル・ビューを生成するのに必要なデータが含まれることを特徴とする請求項1に記載のコンピュータ・システム。

【請求項3】 動作の集合ビューを定義するメニュー定義から前記動作の集合ビューを構築する構築手段をさらに備え、前記メニュー定義は前記集合ビューによって表現される動作オブジェクトおよび（または）他のメニュー定義を識別する項目を有し、さらに、前記構築手段は前記集合ビューを構築するために前記動作オブジェクトおよび（または）前記他のメニュー定義を参照するように配列されていることを特徴とする請求項1または2に記載のコンピュータ・システム。

【請求項4】 前記メニュー定義は、いつ集合ビューがユーザによる選択に対して表示またはイネーブルされなければならないかを定めるコンテキスト情報をさらに含むことを特徴とする請求項3に記載のコンピュータ・システム。

【請求項5】 前記動作オブジェクトに定義された前記識別子は、前記動作オブジェクトの固有識別子と、前記動作オブジェクトの選択に対して前記所望の操作を呼び出す呼び出し機構とを持つことを特徴とする請求項1ないし3のいずれか一項に記載のコンピュータ・システム。

【請求項6】 使用可能な動作のリストを提供するために、前記表示装置に表示されることが可能な動作パレット・ビューをさらに有することを特徴とする請求項2な

いし5のいずれか一項に記載のコンピュータ・システム。

【請求項7】 ユーザ・インタフェースを有し、かつ該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、さらに前記動作は前記ユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システムを操作する方法であって、

(a) 動作を表現するために、前記動作を表現するのに用いることが可能な前記一以上のビューの各々について、該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと、前記動作の選択によって呼び出される操作を識別する識別子とを提供し、さらに前記識別手段に回答して前記所望の操作を呼び出すコマンドを生成するために前記動作オブジェクトによって前記識別子が使用されるステップと、

(b) ビューのユーザ選択がいつ行われたかを識別するステップと、

(c) ビューのユーザ選択の識別に回答して、前記選択に対応する動作オブジェクトを伝えることによって、所望の操作を呼び出すステップと、

を有することを特徴とするコンピュータ・システムの操作方法。

【請求項8】 前記コンピュータ・システムはグラフィカル・ユーザ・インタフェースを表示する表示装置に接続可能であり、また前記動作を表現するのに使用される前記ビューの少なくとも一つは、前記グラフィカル・ユーザ・インタフェースを介して表示可能なビジュアル・ビューであり、さらに前記ビューの少なくとも一つに対する前記動作オブジェクトに定義された前記属性に該ビジュアル・ビューを生成するのに必要なデータが含まれることを特徴とする請求項7に記載のコンピュータ・システムの操作方法。

【請求項9】 動作の集合ビューを定義するメニュー定義から前記動作の集合ビューを構築するステップをさらに有し、前記メニュー定義は前記集合ビューによって表現される動作オブジェクトおよび（または）他のメニュー定義を識別する項目を有し、さらに、前記構築手段は前記集合ビューを構築するために前記動作オブジェクトおよび（または）前記他のメニュー定義を参照するように配列されていることを特徴とする請求項7に記載のコンピュータ・システムの操作方法。

【請求項10】 前記メニュー定義は、いつ集合ビューがユーザによる選択に対して表示または使用可能とならなければならないかを定めるコンテキスト情報をさらに含むことを特徴とする請求項9に記載のコンピュータ・システムの操作方法。

【請求項11】 使用可能な動作のリストを提供するために、前記表示装置に表示されることが可能な動作パレット・ビューをさらに有することを特徴とする請求項7な

いし10のいずれか一項に記載のコンピュータ・システムの操作方法。

【請求項12】アプリケーションを実行するためのプロセッサと、前記アプリケーションによって使用されるデータを格納するメモリと、ユーザ・インタフェースとを備え、該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、また前記動作は前記ユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システムに使用されるプログラム・プロダクトであって、

動作を表現するために前記動作を表現するのに用いることが可能な前記一以上のビューの各々について該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと前記動作の選択によって呼び出される操作を識別する識別子とを提供する手段と、

ビューのユーザ選択がいつ生じたかを識別し、前記選択の対応動作オブジェクトを伝える識別手段とを備え、さらに、

前記識別子を使用して前記所望の操作を呼び出すコマンドを生成するために前記動作オブジェクトは前記識別手段に応答することを特徴とするプログラム・プロダクト。

【請求項13】ユーザ・インタフェースを備え、かつ該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、また前記動作は前記ユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システム上で使用されることを目的としたアプリケーションを開発するためにコンピュータ・システム上で使用され、さらに前記アプリケーションを開発するために前記コンピュータ・システムにロードされた場合に以下の機能を提供するように設けられたプログラム・プロダクトであって、

動作を表現するために前記動作を表現するのに用いることが可能な前記一以上のビューの各々について該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと前記動作の選択によって呼び出される操作を識別する識別子とを生成する手段とを備え、さらに、

前記識別子を使用して前記所望の操作を呼び出すコマンドを生成するために前記動作オブジェクトは前記識別手段に応答することを特徴とするプログラム・プロダクト。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般にユーザ・インタフェースを有し、該ユーザ・インタフェースを介することによってユーザはコンピュータ・システムに実行させる操作を選択することが可能なコンピュータ・システムに関する。また、本発明は特にユーザがそのような

操作を呼び出すことが容易となるようにするメカニズムに関する。

【0002】

【従来の技術】ユーザ・インタフェースは、コンピュータ・ユーザとコンピュータ・システムとの交わりおよび対話を可能とする。ユーザ・インタフェースは、一般にディスプレイ・スクリーンと、キーボード、マウス、マイクロフォン、ライト・ペン等のユーザが制御する入力装置とを実装する。ディスプレイ・スクリーンによって情報およびデータがユーザに対して表示され、またユーザは入力装置を用いてコンピュータ・システムに命令を与えたり、情報を提供する。

【0003】近年、日常業務にコンピュータの力を用いることを希望し、かつ必要とする人々がますます増えている。しかし、一般にユーザは独特のコマンド、演算子、構文の規則等を知る必要があることを望まず、またそのようなレベルの知識が必要とされることを軽減するユーザ・インタフェースの開発に数多くの研究が費やされている。現在もっとも一般的なユーザ・インタフェースの形は、いわゆるグラフィカル・ユーザ・インタフェース（またはGUI）である。概してGUIはユーザにウィンドウおよびアイコンを表示する。ウィンドウには一般にメニュー・バー、ツール・バー、およびクライアント・エリアが含まれる。このクライアント・エリアはいくつかのアイコンを含むものであってもよく、これらのアイコンはユーザが作業を行う実体（アプリケーション、ホルダー、その他）を小さな形で表したものである。

【0004】ユーザがコンピュータ・システム上で特定の操作を呼び出すことが容易となるように、ユーザ・インタフェースはユーザに対して選択を行うためのメカニズムを提供する必要がある。典型的なユーザ・インタフェース環境では、このことは“動作”を定義することによって達成されよう。ユーザ・インタフェースを通じて提供される動作の“ビュー”を介して選択することができ、本発明の目的を達成するために、操作を呼び出すユーザ・インタフェース・イベントとして動作を定義することができる。ユーザ・インタフェース内で動作を示すのに使われる“ビュー”は、視覚的および非視覚的ないくつかの異なる形態をとることも可能である。

【0005】GUIに用いられる視覚的ビューの典型的な例としては、メニュー・バー上の単語または語句である（例えば、図5に示すように、“印刷”動作がワード・ビューから選択される）、ツール・バー上のアイコンまたはビットマップ等のグラフィック表示（例えば、図6に示すように、“印刷”動作が印刷ビットマップ・ビューから選択される）、またはユーザによる選択あるいは入力を可能とする他のビジュアル・コントロールが挙げられよう。そのような他のコントロールの例としては、フォントの選択を可能とするコンビネーション・ボックス

・コントロール（例えば、図7に示す）、またはディレクトリの名前の設定またはツール・バーの名前を設定する等の文字列特性の設定を可能とする項目フィールド

（例えば、図8に示す）が挙げられる。これらのコントロールは、ツール・バーまたはウィンドウ・レイアウトを含むユーザ・インタフェース内のどこにでも現れることができる。視覚的ビューの別の例を図9に示す。この図では、ウィンドウ領域上に位置させる一方でマウスの右ボタンを押し下げることによって表示されるコンテキストから印刷動作を選択することが可能である。また、図10では、印刷動作はウィンドウ・レイアウトにあるプッシュ・ボタン・ビューから選択可能である。

【0006】非視覚的ビューの例としては、印字動作を選択するのに使われる“ctrl+P”などのアクセラレータ・データ、スピーチ入力動作を選択したかどうかを判断するのに使われるスピーチ・パターン・データ、または入力ジェスチャ（例えば、タブレット上でのペンのストローク）が特定の動作の選択であるかどうかを決定するジェスチャ・データが挙げられる。動作を選択するそのようなメカニズムのすべてが“ビュー”と呼ばれる（視覚的または非視覚的であるかどうかは問わない）。

【0007】現在、アプリケーションを開発する場合、ユーザによる動作の選択を可能とするのに必要とされる機能性を付与するために、開発者は多大な努力を払わなければならない。また、ユーザが同一であると機能的に認める動作のビューを表示するアプリケーション間でのユーザ・インタフェースおよび機能的一貫性を達成する上で問題がある。

【0008】それらの動作を可能とさせることが開発者にとって多大な時間の浪費となる理由は、ユーザが同一動作を選択することが可能な場所が一つのアプリケーション中にいくつかの異なる場所として設けられているためである。見てすぐ分かる動作に対するビューは、メニュー・バー、ツール・バー、およびコンテキスト・メニュー上にある。しかし、ユーザに対してダイアログまたはウィンドウを開くプッシュ・ボタンが呈示されている場所もアプリケーション中に多数ある。この“動作”は他の環境下でメニュー・バーまたはコンテキスト・バーからかなり容易に利用できるかもしれない。

【0009】いくつかの動作は、ファイル・オープン・ダイアログのようなダイアログを開く。さらにいくつかの動作はダイアログを開かないけれども操作を実行する。カット、コピー、およびペースト動作のように操作を実行する。それらの操作を結びつけるものは、動作のビューからユーザ利用可能となる事実である。この一例としては、ユーザがメニュー・バーのプル・ダウン、ツール・バー上のツール・バー・ボタン、コンテキスト・メニューの項目、またはダイアログ上の例えばプッシュ・ボタンから印刷オプションを選択することができる場合が挙げられる。特に、印刷ダイアログは、本質的に同

一操作で開かれるけれども、同一結果を達成するのに異なった視覚的選択メカニズムを使用する。

【0010】第2の問題点は、開発者に求められる努力の度合いが増すということである。同一動作を選択する際にユーザによって使用される入力メカニズムには、いくつかの異なるものが存在するというものである。マウスの標準的なポイントおよびクリックのメカニズムはよく知られており、GUIを介して表示された視覚的・ビューを選択するのに使用される典型的なメカニズムである。しかし、動作のいくつかは、キーボード上の特定の文字（例えば、“印刷”動作を選択するための“P”）を介した選択を可能とするアクセラレータ・オプションも有する。さらに、スピーチ・イネーブル・アプリケーションはユーザがスピーク・コマンドを使用することを可能とするので、例えばユーザは印刷ダイアログを開くために“印刷”と言うことができる。また、ペン・ジェスチャが印刷ダイアログを開くために定義されている場合、ペンが利用可能となることによって、動作を選択するためにユーザが利用可能な他のビューが追加されよう。

【0011】さらに、インタフェースのわずかな挙動が、アプリケーション全体にわたってコードの同一部分を繰り返し記述するほどに広まるという状況がある。この例としては、何ら変更を加えることなしにユーザがウィンドウをキャンセルするのを可能とさせるいくつかのダイアログ・タイプのウィンドウをアプリケーションが有する場合が挙げられる。この機能の視覚的表示としては、例えば“キャンセル”プッシュ・ボタンが挙げられる。しかし、通常は同様の結果を得るキーボード・メカニズムもあり、例えば“Esc”を選択する。従来、開発者はこの機能を達成するために3つのことをしなければならなかった。すなわち、（a）ダイアログ上にプッシュ・ボタンを設けること、（b）アクセラレータ・テーブルにエスケープ・アクセラレータを加えること、（c）アプリケーション・コード中にキャンセル手順を加えることである。これらの3つの段階は、常に個々のウィンドウでエスケープ機能が必要とされる場合にいつも繰り返されなければならない。

【0012】したがって、アプリケーションの開発者は、現在のところそれらの異なるシナリオに対処するために膨大な量のコードを記述しなければならず、またスピーチ等の新しい技術が導入された場合、ユーザが使用可能な新規の選択メカニズムを作るための追加の仕事が開発者に必要となる。開発者がおぼえなければならない順列がいくつかあるので、ある種の選択メカニズムがプロダクト内で使用可能とはならない場合がある。このことは、一つのアプリケーションは一つの方法で働き、他方では働かないという製品間を越えた使用適正上の問題につながる。このことに加えて、同一動作上の種々のビューは一般にいくつかの異なる方法でもって構築され、また一度構築されるとユーザ不要のカスタマイズ機能を

ほとんどもない。

【0013】

【発明が解決しようとする課題】以上説明したように、現在、動作に関する種々の視覚的および非視覚的ビューを完全に支持するために必要とされる開発者が行うべき仕事は数多くあることは明かである。したがって、本発明は上記問題を軽減するコンピュータ・システムおよび方法を提供することを目的とする。

【0014】

【課題を解決するための手段】第1の態様によれば、本発明は、ユーザ・インタフェースを持つコンピュータ・システムを提供する。このコンピュータ・システムは、ユーザ・インタフェースを介してユーザが所望の操作を呼び出す動作を選択することが可能であり、またユーザ・インタフェースを介して一以上のビューによって動作の選択が可能であるコンピュータ・システムであって、動作を表現するために、動作を表現するのに用いることが可能な一以上のビューの各々について、該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと、動作の選択によって呼び出される操作を識別する識別子とを提供する手段と、ビューのユーザ選択がいつ生じたかを識別し、選択の対応動作オブジェクトを伝える識別手段とを備え、さらに動作オブジェクトは識別手段に応答して所望の操作を呼び出すコマンドを生成するために識別子を使用することを特徴とする。

【0015】好ましい実施形態例では、コンピュータ・システムはグラフィカル・ユーザ・インタフェースを表示する表示装置に接続可能であり、また動作を表現するのに使用されるビューの少なくとも一つは、グラフィカル・ユーザ・インタフェースを介して表示可能な視覚的ビューであり、さらにビューの少なくとも一つに対する動作オブジェクトに定義された属性に該視覚的ビューを生成するのに必要なデータが含まれる。

【0016】第2の態様によれば、本発明はコンピュータ・システムの操作方法を提供する。この方法は、ユーザ・インタフェースを有し、かつ該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、さらに動作はユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システムを操作する方法であって、(a)動作を表現するために、動作を表現するのに用いることが可能な一以上のビューの各々について、該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと、動作の選択によって呼び出される操作を識別する識別子とを提供し、さらに識別手段に応答して所望の操作を呼び出すコマンドを生成するために動作オブジェクトによって識別子が使用されるステップと、(b)ビューのユーザ選択がいつ生じたかを識別するステップと、(c)ビューのユーザ選択の識別にตอบสนองして、選択の対応動作オブジェクトを伝えることによ

て、所望の操作を呼び出すステップとを有することを特徴とする。

【0017】第3の態様によれば、本発明はプログラム・プロダクトを提供する。このプログラム・プロダクトは、アプリケーションを実行するためのプロセッサと、アプリケーションによって使用されるデータを格納するメモリと、ユーザ・インタフェースとを備え、該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、また動作はユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システムに使用されるプログラム・プロダクトであって、動作を表現するために、動作を表現するのに用いることが可能な一以上のビューの各々について、該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと、動作の選択によって呼び出される操作を識別する識別子とを提供する手段と、ビューのユーザ選択がいつ生じたかを識別し、選択の対応動作オブジェクトを伝える識別手段とを備え、さらに動作オブジェクトは識別手段に応答して所望の操作を呼び出すコマンドを生成するために識別子を使用することを特徴とする。

【0018】第4の態様によれば、本発明はユーザ・インタフェースを備え、かつ該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、また動作はユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システム上で使用されることを目的としたアプリケーションを開発するためにコンピュータ・システム上で使用され、さらにアプリケーションを開発するためにコンピュータ・システムにロードされた場合に以下の機能を提供するように設けられたプログラム・プロダクトであって、動作を表現するために、動作を表現するのに用いることが可能な一以上のビューの各々について、該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと、動作の選択によって呼び出される操作を識別する識別子とを生成する手段を備え、さらに、動作オブジェクトは識別手段に応答して所望の操作を呼び出すコマンドを生成するために識別子を使用することを特徴とする。

【0019】本発明によれば、動作を取り扱うために単一のメカニズムを設ける。このメカニズムはOOPオブジェクトの形態を取るもので、該OOPメカニズムは上記動作を表現するのに用いられる使用可能なビューの各々について、該ビューを提供するのに必要とされる属性、および動作を選択することによって呼び出される操作を識別する識別子を定義する。

【0020】オブジェクト指向プログラミング(OOP)は、“オブジェクト”へ送られた“メッセージ”によって必要とされる機能を実現するソフトウェア開発に対する特定のアプローチである。“オブジェクト”は、データ

の集合および関連するソフトウェアの手順（しばしば“方法”と呼ばれる）を含むソフトウェア・パッケージである。一般にオブジェクトは特定の“オブジェクト・クラス”のインスタンスとして生成され、該クラスは手順を定義するテンプレートである。“メッセージ”はオブジェクトへ送られる信号であり、該オブジェクトを要求してその手順の一つを実行する。したがって、オブジェクトに送られたメッセージは、要求された機能を実現するために呼び出される手順をもたらすであろう。

【0021】本発明の目的のために、動作を扱うために与えられたOOPオブジェクトを“動作オブジェクト”と呼ぶことにする。好ましい実施形態例では、この動作オブジェクトは、動作を表現するのに用いられる使用可能なユーザ選択メカニズム（または“ビュー”）の各々に対して、（a）視覚的ユーザ選択する動作の視覚的表現構築、または非視覚的メカニズムを介して選択を可能にさせる他動作の視覚的表現構築のいずれかに必要とされる属性と、（b）動作の選択によって呼び出される操作を識別する識別子とを定義する。好ましくは、この識別子は選択された動作オブジェクトの単一識別子と、実行する所望の操作をもたらすように動作が選択された場合に自動的に呼び出されるアプリケーション呼び出しメカニズムとからなる。そのようなアプリケーションによって、どの入力メカニズムが呼び出しを生じたか（例えば、マウス、キーボード、スピーチ入力等）または動作が選択されたのはどのビューからかに気がつくことなしに動作が選択された場合に定義される単一の入口点、アプリケーションによって定義することができる。

【0022】さらに、好ましい実施形態例では、動作オブジェクトは該動作の現在の状態、例えば使用可能か、使用不可能か、ラッチされてるか、または定義されているかどうかを定義する。動作の状態は、関連する視覚的動作上のそれ自身の見える状態を定義するとともに、動作が非視覚的動作から選択可能であるかどうかを記述する。使用可能となった状態は、動作が現在のところ使用可能であるけれども選択されていないことを示す。ラッチされた状態は動作がラッチ可能であり、したがってラッチされた状態を明確に選択されることなしにアプリケーション内でさらなる相互作用に適用されることができる。この一例として、ワードプロセッサにおける“太字”動作が挙げられる。この動作がラッチされたとなると、ユーザがデータをさらに入力することによって新規テキストに加えられる太字属性となる。使用不可能な状態は、動作が現在のところ使用されていないことを示す。未定義の状態は動作が現在のところ使用されていないことを示す。この一例としてワードプロセッサにおいては、各々が異なるフォントを持ついくつかのセンテンスをいつ選択するかである。“フォントの選択”動作は常にフォント選択ビュー内に現在表示されたフォントの視覚的フィードバックを提供する。しかし、この場合、動作

はユーザに与えられるべき適切なフィードバックを決定することができないことから、動作の状態は未定義なものとなる。

【0023】この機能性が一般に与えられている従来のシステムの個々の選択および入力メカニズムから該機能性を取り除くことによって、単一動作オブジェクトはすべてのビューを知らせる責任を取ることができ、それらのビューから任意の変更を状態に加えることが現在可能となる。これによって、アプリケーション・コードからビュー同時実行論理の一部が取り除かれる。オブジェクト間の通知フレームワークに関する標準的な技術を提供するオペレーションは一般に当業者に知られている。例えば、IBMのVisual Age C++プロダクトでは、この機能を実行するクラスが提供される。他のクラスは、観察者および通知者クラスから引き継がれる標準的なOO技術を用いることによってこの挙動をピックアップすることができる。

【0024】GUI環境では、動作の集合に対する選択メカニズムがいくつかの方法でしばしばひとまとめにされる。一般的に、動作の集合はメニュー・バー、ツール・バー、およびポップ・アップ・メニューとして視覚的に表わされる。しかし、非視覚的な表現もまた提供することが可能であり、例えば動作の集合に対してキーボード選択を可能とさせるアクセラレータ・テーブルが挙げられる。一例として、ツール・バーはいくつかのグラフィック・ボタンを有するものであってもよく、各々のボタンが特定の動作のビューとなる。一つのボタンを選択することによって、該ボタンが所望の操作を呼び起こす特定の動作が引き起こされる。本発明の目的を達成するために、そのようなビューのグループ化を該グループ化が視覚的（例えば、メニュー・バー、ツール・バー）または非視覚的（例えば、アクセラレータ・テーブル）であろうとなかろうとかかわりなく、本明細書ではそのようなグループ化を“集合ビュー”と呼ぶこととする。

【0025】ここで言及すべきことは、集合ビューは他の集合ビューに組み入れられてもよいということである。例えば、いくつかの異なる動作が含まれるという点で、メニュー・バーを集合ビューとしてみなすことができる。メニュー・バーのいくつかの項目（例：“ファイル”という語）はそれ自身で集合ビューを表現してもよい。なぜなら、ひとたび選択されると他の集合ビューが消えてもよいからである。例えば、“ファイル”という語を選択すると、集合ビューは消えてリストのかたちになるようにしてもよい。このリストに含まれる語は特定動作のビューである。例えば、“開く”はファイルを開く際に用いられる“開く動作”のビューである。

【0026】本発明の好ましい実施形態例では、集合ビューの構築を扱う単一のメカニズムが与えられる。このメカニズムは、メニュー・バーに言及したOOPオブジェクトのかたちをとるもので、（a）動作オブジェク

ト、分離文字、またはメニュー定義の他のインスタンスに対する参照のいずれかを個別に表わす項目の集合と、

(b) 動作上の集合ビュー (視覚的であるか非視覚的であるかは問わない) の表現を提供するのに必要とされる属性とが含まれる。好ましくは、メニュー定義は動作の集合ビューが表示されるか、あるいは選択可能となるコンテキストも定義する。

【0027】本発明の好ましい実施形態例では、コンピュータ・システムは構築手段を有する。この構築手段は、集合ビューを定義するメニュー定義から動作の集合ビューを構築する。また、メニュー定義は動作オブジェクトおよび (または) 集合ビューによって表わされる他のメニュー定義を識別する項目を有する。さらに、構築手段は集合ビューを構築するために、それらの動作オブジェクトおよび (または) 他のメニュー定義を参照するように配列されている。

【0028】メニュー定義は動作オブジェクトを参照し、動作のビューを構築するのに必要なすべての属性は動作オブジェクトに含まれるので、メニュー定義を複数の特定の集合ビューに用いることができる。例えば、特定のメニュー定義はテキスト状のメニュー・バー、あるいはそれに相当するグラフィック・ツール・バーを構築するのに使用される。なぜなら、どちらの場合でも集合ビューの特定項目に対する動作オブジェクトが参照され、該動作オブジェクトはどのようなタイプの集合ビューが構築されているかに応じて適当なテキスト (例えば、"印刷") またはアイコン (例: プリンタを表わす記号) のいずれかを設けることができる。

【0029】すでに述べたように、好ましい実施形態例において、メニュー定義には該メニュー定義によって表わされる集合ビューを表示または使用可能とすべき環境を識別するコンテキスト情報が含まれる。省略時解釈によって、またそれによって関連した"集合ビュー"によって、メニュー定義はいつでも使用可能または表示される。しかし、一例としてポップ・アップ・メニューまたはコンテキスト・メニューは、それに関連した少なくとも一つのコンテキストを持つことによって、それ自身のコンテキストが真である場合にのみ表示されるようにすることが可能である。

【0030】コンテキストは、任意のメニュー定義または集合ビューに対して適用可能であってもよい。例えば、アプリケーションのコンテキスト (例えば、"ファイル"プル・ダウン・メニュー、"編集"プル・ダウン・メニュー等) を無視してメニュー・バーは常に表示される埋め込み型メニュー定義と、特定のコンテキストに対してのみ適用可能ないくつかの項目とを含むものであってもよい。具体例として、メニュー・バーはグラフィック画像を処理する動作の集合を与える埋め込み型メニュー定義を含むものであってもよい。この動作の集合はグラフィック・オブジェクトがユーザによって選択された

場合のみ適用可能なものであってもよく、それによってグラフィック画像が選択されない場合に、メニュー・バーは埋め込み型メニュー定義を表示しなくすむ。メニュー・バーは、現在のアプリケーション・コンテキストに対する埋め込み型メニュー定義のコンテキストを含むことによってその決定を下すことができる。

【0031】好ましい実施形態例では、使用可能な動作のリストを提供するために、表示装置 (55) に表示されることが可能な動作パレット・ビュー (280) をさらに有する。これによって、ユーザは動作の属性を直接編集することが可能となる。さらに、パレット・ビューは、好ましくはユーザが視覚的集合ビューへの動作を容易に加えることが可能となるドラッグ・アンド・ドロップ対応する。

#### 【0032】

【発明の実施の形態】本発明にもとづくシステムの好ましい実施形態の一例を、該システムの主要な機能的構成要素を示す断面図である図1を参照しながら説明する。図1において、典型的なコンピュータ・システムは、該システムにインストールされたオペレーティング・システム1、中央処理装置2、メモリ15、システムの一部として内蔵またはリモート・アクセスされる記憶装置、およびコンピュータ・システムに接続可能な表示装置を備える。

【0033】中央処理装置2は、表示装置55上に表示を行う表示手段50に対して表示データを送るように設けられている。当業者によって容易に理解されるように、当該技術分野において表示手段の操作は一般に知られているので、これ以上詳細に説明する必要はない。それが表示装置55へデータを送るのに使用される種々のソフトウェアおよびハードウェア・コンポーネントを包含することを言えば十分である。IBMのOS/2オペレーティング・システムでは、プレゼンテーション・マネージャーと呼ばれるOS/2のレイヤーがこの目的のために用いられる。

【0034】本発明の好ましい実施形態例では、標準的なGUIが表示装置55上に表示されていくつかのウィンドウおよびアイコンがユーザに対して呈示される。ソフトウェア・アプリケーション7はGUIのなかの一つ以上のウィンドウに連携している。特定の操作を呼び出すように、ユーザがアプリケーション7と対話できるようにするために、GUIによって好ましくはユーザが選択できるある種の動作の視覚的ビューを提供する。動作中に、ある種の非ビジュアル・ビュー、例えばアクセラレータ内のアクセラレータ・データ等によって、ユーザがキーボードの記号を選択して動作を呼び出し、それによって対応する操作を呼び出すようにしてもよい。

【0035】ユーザはコンピュータ・システム1に接続された適当な入力装置によってビューを選択する。そのような入力装置の例としては、マウス70、キーボード

80、マイクロホン90、またはペン100が挙げられる。ボタン、メニュー・バー、ツール・バー等の視覚的ビューに関して、当業者ならば容易に分かることではあるが、ユーザは一般的にマウスを選択する。しかし、ユーザはキーボード80上のキー入力（例えばアクセラレータ・キー）による動作の選択、ペンによるジェスチャー入力、またはマイクロホンによる音声入力を選ぶこともできよう。しかし、入力信号は発生すると、該入力信号は入力手段3によって受信され、分析のために識別手段へ送られる。当業者が容易に理解できるように、入力手段3の機能は一般にオペレーティング・システム／該システムのGUIによって提供される。例えば、入力手段がマウスである場合はマウス・ドライバによって与えられる。

【0036】識別手段4は、好ましくはプロセッサ2の制御したで実行されるソフトウェア・プロセスであり、入力信号がある動作の特定のビューを選択することと一致するかどうかを認識する。好ましい実施形態例では、この識別の実行の仕方は動作フレームワークの部分として与えられるもので、図2ないし図4を参照してより詳細に後述する。しかし、この機能を動作フレームワークの部分として提供する必要はなく、その代わり、当業者に容易に理解されるように識別手段4を他の所へ設けることができる。

【0037】ユーザによるビューの選択が行われたことを識別手段4が識別すると、メモリ15内の関連オブジェクト20が知らされる。つぎに、プロセッサ4の制御下で、そのオブジェクトは該オブジェクトによって定義される特定の動作が呼び出される。例えば、ビューが印刷動作オブジェクトに対応するならば、印刷動作オブジェクトは印刷動作の選択によって呼び出される印刷操作を定義する。メモリ15内にオブジェクトが生成されるように、生成手段5を設ける。これによってアプリケーション開発者がアプリケーション7によって用いられるオブジェクトを定義することが可能となる。本実施形態例では、この機能は動作フレームワークの部分として提供されるものであり、図2および図3を参照してより詳細に後述する。しかし、動作フレームは必須のものではない。当業者が容易に理解するように、それに代えて開発者は開発者自身の標準OOP技術を用いたオブジェクトを構築することができる。クラスは該クラスに属するオブジェクトに関連した標準的なビュー属性および手順を提供する。このオブジェクトのクラスを定義することによって、開発者はそのクラスから直接的に多くのインスタンスを生成することができる。

【0038】好ましい実施形態例のシステムは、プロセッサ2の制御下で、メモリ16内に格納されたメニュー定義オブジェクト20から一括したビューを生成するのに用いられる構成手段6を有する。メニュー定義オブジェクトは、動作オブジェクトと同様に、生成手段5によ

って生成され、また構成手段6はツール・バー、メニュー・バー、およびアクセラレータ・テーブル等の集合ビューをそれらのビューから生成するのに用いられる。好ましい実施形態例では、構成手段6もまた動作フレームの部分となすが、この機能を他の所に設けることが可能であることは当業者によって容易に理解されよう。必要とされるもののすべては、メニュー定義オブジェクトの定義から集合ビューを構築するいくつかの機能を提供することである。集合ビューを構築するためのプロセスを図3を参照しながらより詳細に後述する。

【0039】好ましい実施形態例のシステムを図2を参照しながらよりいっそう詳細に説明する。図2はシステムの主要な機能的構成要素を示す断面図である。すでに述べたように、好ましい実施形態例では、システムは動作フレーム10を備える。この動作フレーム10はタスク実行のためのアプリケーション・コードとともに論理的に"配線"することが可能な一組のベース・オブジェクトを含む。動作フレーム10は、オペレーション・システムの基本的なファイル・システム30を用いて、フレームワーク10内に存在するオブジェクトのビューに対応付けられたビュー属性をシステム記憶装置からロードし、かつメモリ35に格納する。

【0040】特定のアプリケーションによって要求されたメニュー定義および動作オブジェクトが一度構築されると、オペレーティング・システム（IBSのOS/2オペレーティング・システムによって提供される）の表示処理システム40を用いてディスプレイ・インタフェース50に必要な信号を生成し、ディスプレイ装置55上にそれらのオブジェクトのビューを表示する。一般に、それらのビューはGUIウィンドウ化インタフェース60に表示される。しかし、スピーチ等の新規入力および出力機構を導入することで、スピーチ出力装置等の他の出力装置をディスプレイ装置55に加えて、あるいはその代わりとして用いることが可能であることが理解されよう。

【0041】選択候補から選択するために、ユーザはGUIインタフェース60を介してシステムと対話する。ユーザは複数の方法で選択を行うことができる。それらの方法としては、マウス70等の標準的なポインタ装置、キーボード80、スピーチ入力装置90、ペン入力装置100、またはタッチ・スクリーンを介することが挙げられる。どのような手段でユーザが選択を行おうとも、種々のデバイス・ドライバがオペレーティング・システムのイベント・マネージメント・システム110によって処理されるイベントが生成されよう。IBMのOS/2オペレーティング・システムでは、イベント・マネージメント機能は標準制御およびウィンドウ化メッセージ化システムが与えられることによって該オペレーティング・システムおよびプレゼンテーション・マネージャによって処理される。



【0042】後で詳細に説明するように、選択イベントは動作フレームワーク10内でオブジェクト20の一つに対応付けられる。これによって、どのようにして、あるいはどこで選択イベントが生成されるかが問われることなく要求タスクが実行されよう。

【0043】動作フレームワーク10内の基本オブジェクトについて図3を参照しながら説明する。図3は、本発明の好ましい実施形態例に従って、どのようにオブジェクトが構成されることによって多数のビューが可能となるか、さらにどのようにしてオブジェクト間の相互関係を通じてどのようにして集合オブジェクトが生成できるかについて説明するオブジェクトのブロック図である。

【0044】好ましい実施形態例では、特定の動作に対応付けられたビュー属性のすべてを含む基本動作オブジェクト200が生成される。好ましい実施形態例では、動作フレーム10は、根底にあるファイル・システム30に格納される共通の動作タイプ210についてのビュー属性212からなるアプリケーション拡張可能な所定の集合を提供する。この動作タイプの単一集合は、システム内の2つ以上のオブジェクトまたはアプリケーションによってアクセス可能であり、その結果動作タイプ・データの個別化はそれら共通の動作タイプを使用するいずれのオブジェクトまたはアプリケーションにたいしても影響を及ぼす。

【0045】共通動作タイプは、“カット”、“コピー”、“ペースト”、および“印刷”のようなタスクを含み、また各動作タイプについて動作のビューを表示するのに必要なビュー属性のすべてが格納されている。そのようなビュー属性は一般に以下のような情報を含む。

- ・メニュー・テキストおよびボタン・テキスト（例えば、“印刷”）
- ・グラフィック（例えば、印刷ビットマップまたはアイコン）
- ・アクセラレータ・データ（例えば、“Ctrl + p”）
- ・ニーマニックス（例えば、メニュー移動に使用される“P”）
- ・ホバー・ヘルプ（例えば、“選択されたオブジェクトを印刷”）
- ・スタイル属性（動作が動作プレート（利用可能な動作のリスト）内でユーザに表示されるか、またはラッチ可能であるかどうか）
- ・カテゴリ（どのようなタイプの動作であるか。例えば、ファイル動作）
- ・スピーチ・パターン（定義した場合）
- ・ジェスチャ（定義した場合）

【0046】動作タイプ属性212は動作オブジェクト200上にインスタンスを構築するのに用いられるので、このデータは特定の動作タイプに開かれているクラ

スとして表示することができる。このことは、たとえばアプリケーションがいくつかの印刷動作オブジェクト200、例えば異なるプリンタで印刷することを表示するものを持つとともに、“印刷”動作タイプ210の一組のデータのみがあるであろうことを意味する。

【0047】動作タイプ属性に加えて、動作オブジェクト200自体はある動作の特定のインスタンスに関係する。例えば、各動作オブジェクトは以下の通りである。

・個別識別子214（ユーザ選択からのイベント経路指定に使用）

・現在の状態216（動作インスタンスが使用可能であるか、ラッチされているか、使用不可能であるか、あるいは未定義の状態であるか）

・呼び戻し機能218（動作の選択結果として呼ばれるであろうアプリケーション定義コードを識別）

【0048】好ましい実施形態例では、“印刷”動作等の共通の動作を生成することを望む開発者は、印刷オペレーション220の生成のみを必要とし、動作オブジェクトに対するこの動作にポインタ218を定める。つづいて、動作が選択されると印刷オペレーションが呼び出される。印刷オペレーションは、動作タイプ・レベルで与えられ、システム内に2つ以上の印刷動作オブジェクトが存在すると、選択された場合にそれら2つ以上の印刷動作オブジェクトすべてが同一印刷機能220を呼び出す。あるいは、印刷機能をインスタンス・レベルでもって特定することができる。この場合、各印刷動作インスタンスに対して異なる呼び出し機能が存在するであろう。

【0049】動作タイプ属性212が動作オブジェクト200を介して設定され、かつ照会される。したがって、アクション・オブジェクトは特定の“方法”を支持して特定の機能を実行する能力を提供する。ここで特定の機能とは、以下の機能である。

・動作オブジェクトの属性を設定し、かつ照会する。

・動作の現在の状態、すなわち動作が強調表示されているか、使用可能か、あるいはラッチされているかどうかを保持する。

・任意の属性が変更された場合に任意の“オブザーバー”を通知すること—すべてのビュー属性が同一の場所で保持され、かつ更新されたとすると、動作の属性または状態が変わる場合に動作オブジェクトは現在指令を表示している任意のビューを通知することができ、それによって表示をリフレッシュすることができる。

・動作に対して保持されるビュー定義の異なるタイプ、すなわちツール・バー・ビュー、レイアウト・ビューを許可する。

・生成される個別イベント識別子を与え、それによって動作に対するすべての選択機構および入力機構は、アプリケーション・イレスペクティブに戻るか、どのようにユーザが呼び出しを生成したか一つのイベントを生成す

る。

・動作属性を保存および復元する。

【0050】上記の記載から、そのようなオブジェクトを設けることによって、共通に定義された視覚的または非比較的ビュー属性のすべてを保持した共通動作の生成が可能であることが明らかであろう。動作オブジェクトは選択された動作によって実行される実際のオペレーションを扱わないので、アプリケーションは容易にビュー選択機構をピック・アップすることができ、さらにアプリケーション定義域に適用できる実際のオペレーションは該アプリケーションによって定義される。

【0051】上記のアプローチは、ユーザによって選択可能な動作を可能とさせる上で開発者側に必要とされる仕事の量を緩和する。さらに、動作上の異なるビュー間のビュー同時並行性を保つのに常に必要と思われる大量の仕事を取り除くことから、異なるアプリケーション間の共通動作のインタフェースおよび機能性にかかわる矛盾を伴う問題点の大部分を解決する。このことのすべてが、動作オブジェクトと動作ビューとの間で気をつけられる。動作レベルで定義された動作についての任意のビューがあるとすると、動作の属性または状態が変化した際に通知されるビューを可能とする動作オブジェクトのオブザーバとして登録される。つづいて、アプリケーション定義ビューに到達して必要ならばその表示を変更する。

【0052】上記の記載では、動作オブジェクトがどのように構築され、かつ容易に生成されるそのようなオブジェクトの一致したビューを可能とさせるかについて論じている。つぎに、好ましい実施形態例のシステムが集合ビューを取り扱う方法を検討する。生成される集合ビューを可能とするために、また一方でインタフェース一貫性および選択一貫性を保つために、動作インタフェースからなる群を含むツリー状構造を生成する。一様な集合からなるメニュー項目232を保持することによってその構造を達成するメニュー定義オブジェクト230を構築するもので、印刷動作オブジェクト200、表示装置内でのグループ化を区別するのに用いられる分離文字、またはそれ自体が他の一様な組からなるメニュー項目を含む他のメニュー定義インスタンスのような動作インスタンスのいずれかを表示することができる。

【0053】好ましい実施形態例では、基本メニュー定義オブジェクト230が生成され、特定メニュー定義に関連するすべてのビュー属性を含む。好ましくは、動作フレームワーク10は根底にあるファイル・システム30内に格納された共通メニュー定義タイプ240に関する所定の集合からなるビュー属性242を与える。標準動作タイプ定義210とは異なり、メニュー・タイプ定義240は内容としてオブジェクトまたはアプリケーションによってカスタマイズ可能ではなく、またメニュー内での順序付けは多数のアプリケーションをまたがるよ

りも単一のアプリケーションに関係する。したがって、メニュー定義オブジェクトはメニューのカスタマイズ化を提供し、また個々の内容および順序づけをロードしたり、復元するのに必要とされるロードおよび復元機能性も提供される。

【0054】共通のメニュー・タイプには、一般に標準的なメニュー、ファイル・メニュー、編集メニュー、およびヘルプ・メニューが含まれる。各メニュー・タイプに対して、集合ビューを形成するのに必要とされるすべてのビュー属性が存在する。このデータは一般に以下の情報を含む。

・メニュー・テキスト（例えば、“ファイル”）

・グラフィック（例えば、ファイル・ビットマップまたはアイコン）

・ニモニック（例えば、メニュー移動に使用される“F”）

・ホバー・ヘルプ（例えば、“ファイルを操作することが可能な動作の組”）

・コンテキスト（例えば、“いつも”。すなわち、アプリケーションまたはオブジェクト・コンテキストから独立してこのメニューを常に表示する）

・項目（新規動作タイプ、開く動作タイプ、分離文字、上書き保存動作、名前を付けて保存動作タイプ、分離文字、印刷動作タイプ）

【0055】好ましい実施形態例では、開発者は共通のメニュー定義、例えば“ファイル”メニュー定義の生成を望む。この“ファイル”メニュー定義は、ファイル・メニュー・タイプ240のメニュー項目リスト内にある動作タイプを確保することだけを必要とするもので、すでに生成され、かつ呼び出し機能は登録されている。そうでない場合には、動作はメニュー定義インスタンス内のメニュー項目リストに含まれないであろう。このことはある種の利点をもたらすもので、共通メニュー定義が用いられて上記組に含まれる動作を支持する必要のあるアプリケーションなしにアプリケーション・メニューを生成することに用いることができる。

【0056】メニュー・タイプ属性242はメニュー定義オブジェクト230を介して設定され、かつ照会される。したがって、メニュー定義オブジェクト230は一定の“方法”を支持して一定の機能を実施する能力を提供する。例えば、

・任意の属性が変化した場合に任意の“オブザーバ”を通知するメニュー定義オブジェクトの属性を設定および照会すること、

・メニュー定義属性を保存および復元することである。

【0057】標準的な動作およびメニュー・タイプを提供することによって、ユーザ・インタフェースの一貫性が開発生産性として増加する。しかし、当業者が容易に理解できるように、動作オブジェクトおよびメニュー定義の概念は、所定の動作210およびメニュー240タ

イプを用いることなしに実現される。例えば、直接に単一の動作オブジェクト200内にある印刷動作に対するすべてのビュー属性を定義することによって（印刷動作タイプを参照するよりはむしろ）、このことによって生成される印刷動作のマルチプル・ビューおよび支持されるマルチプル選択機構がさらに可能となろう。

【0058】ひとたびメニュー定義インスタンス230が生成されると、標準集合ビューを単一メニュー定義から生成することが可能となる。これらのビューは標準メニュー・バー250、標準ツール・バー200、コンテキスト・メニューまたはポップ・アップ・メニュー、およびアクセラレータ・テーブル270を含む（なお、アクセラレータ・テーブル270は後述する図4の説明でよりいっそう詳細に説明する）。他の標準的な集合ビューにはスピーチ・テーブルおよびジェスチャ・テーブルが含まれよう。

【0059】カスタマイズ化によってユーザはメニュー・テキスト、ツール・バー・グラフィック、ホバー・ヘルプ等のアクションの視覚的属性のいずれかを変更することが可能となる。さらに、好ましい実施形態例では、動作フレームワーク10は動作パレット・ビュー280を提供する。この動作パレット・ビュー280によって、表示可能な属性を持つシステム内のすべての使用可能な動作のリストが与えられる。この標準的な動作パレット・ビュー280は、好ましくはドラッグ・アンド・ドロップ対応であり、ユーザが動作を視覚的集合ビューに加えることを可能とする。また、ユーザが動作の属性の能力を編集することも可能とする。この標準的なビュー280と基本動作オブジェクト200内の通知との間の通知を介して、メニュー定義から生じたすべての集合ビューは変更に応じて新規属性によって自動的に更新されよう。

【0060】また、カスタマイズ化によってユーザは、メニュー・テキスト、ツール・バー・グラフィック、およびホバー・ヘルプのように、メニュー定義230の視覚的属性を変えることが許される。メニュー定義は動作のダイナミックな階層的配列であることから、ユーザは視覚的集合ビュー内の項目を削除および追加することができ、または必要に応じてそれを再はい得ることができる。メニュー属性あるいはコンテキストのいかなるカスタマイズ化も自動的にそのオブジェクト上の任意の集合ビューに自動的に通知される。このことは、メニュー定義上のビュー間のビュー同時実行を行うために開発者に必要とされる仕事の量をさらにまた軽くする。

【0061】実際に必要とされる動作をユーザが誤って削除した場合に対処するために、好ましい実施形態例では、標準的な動作パレット・ビューからアクションの完全な組をいつでも得ることができる。

【0062】メニュー定義においてコンテキスト属性を用いることによって、所定のコンテキストが真である場

合にユーザのみに表示されるメニューを定義することができる。現行のコンテキストは、アプリケーション・ウィンドウまたはクライアント領域に対応付けられており、さらに通常はユーザ選択に関連する。集合ビューが対応付けられたウィンドウまたはクライアント領域（例えば、メニュー・バーはいつもウィンドウに付いている）に現行のコンテキストを加えることによって、メニュー定義はこのウィンドウまたはクライアント領域の属性のオブザーバとなる。それ自身の内部コンテキスト属性が現行のウィンドウまたはクライアント領域のコンテキストと一致しないとすると、それ自身の上のすべてのビューを隠すように通知し、それとは反対にコンテキストが一致するとメニューに対して表示するように通知する。この特徴は任意の集合ビューにおける自動的コンテキスト挙動を可能とする。

【0063】本発明の好ましい実施形態例にもとづくユーザ選択の結果としてタスクを実行する制御の流れを図4を参照しながら説明する。

【0064】ユーザは、グラフィカル・ユーザ・インタフェースを介して動作を選択することを可能とするいくつかの入力メカニズムを持ってもよい。図4は、“印刷”動作を呼び出すための3つの異なる方法が示されている。すなわち、メニュー・バー項目300でのマウスによる選択、ツール・バー・ボタン310でのマウスによる選択、またはキーボード320による“Ctrl+P”のキー組み合わせである。

【0065】例えばメニュー・バーおよびツール・バーの標準的な集合ビューがメニュー定義インスタンスから構築された場合、該集合ビューはメニュー項目のリストを繰り返して該リスト内にある各動作インスタンスのウィンドウを生成する。個別動作識別子214は、動作を表す関連メニュー項目またはツール・バーのウィンドウ識別子に使用される。したがって、ユーザがツール・バー・ボタンまたはメニュー項目のいずれか一方を選択すると、個別動作識別子214が含まれるのでシステム・イベント・ルータ330を介して生成かつ送出されるイベントは呼び出される必要のある動作インスタンスに正確に一致する。イベントの生成はGUI内の標準的なウィンドウ・メッセージ・システムによって与えられる。さらに、システム・イベント・ルータ330は図1に示す識別手段4によって提供される機能を表す。

【0066】アクセラレータ・テーブル270の標準的な集合ビューがメニュー定義から構築される場合、動作インスタンスのメニュー項目リストを繰り返す。リスト内の各動作リストについて、アクセラレータ・テーブル・オブジェクトは動作インスタンスからそれ自身のアクセラレータ・データを要求し、動作個別動作識別子214を持つ復帰キー組み合わせをそれ自身の内部ブック・アップ・テーブルに加える。内部ブック・アップ・テーブルはアクセラレータ・データ275からなるリストであ

る。さらに、ウィンドウIDが格納されて、キーボードを介した動作を可能とするアクティブ・キーストロークの組と該動作が相互作用するGUI内のアクティブ・ウィンドウとの相互関係が得られる。ユーザによってキー組み合わせが押された場合、アクセラレータ・テーブルはルック・アップ・テーブルに対してチェックを行い、動作に対してキー順序が登録されているかどうかを判断する。登録されている場合、順にシステム・イベント・ルータ330を経由して送られ、かつ呼び出される必要のある動作インスタンスに正確に一致する個別動作識別子214を用いるイベントを生成する。さらに、動作200がイベントを受け取った場合、記録されたコール・バック機能220を呼び出してタスクを呼び出す。

【0067】上述したことから、好ましい実施形態例のシステムによって多数のビューから単一イベントを生成できることが明らかであろう。したがって、アプリケーションはそれ自身でユーザがどのようにして呼出しを生成したかについて関わる必要はない。好ましい実施形態例のメカニズムを用いることで、共通の動作ビューは直接動作オブジェクトの属性を問い合わせ、必要なデータすべてを一カ所に保持し、さらに自動的ビュー同時実行によって、アプリケーションの開発者は互いに同調した動作ビューを保つことを心配する必要はもはやなくなる。

【0068】まとめとして、本発明の構成に関して以下の事項を開示する。

(1) ユーザ・インタフェースを有し、かつ該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、さらに前記動作は前記ユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システムであって、動作を表現するために前記動作を表現するのに用いることが可能な前記一以上のビューの各々について該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと前記動作の選択によって呼び出される操作を識別する識別子とを提供する手段と、ユーザによるビューの選択がいつ生じたかを識別し、前記選択に対応する動作オブジェクトを伝える識別手段とを備え、さらに前記識別子を使用して前記所望の操作を呼び出すコマンドを生成するために前記動作オブジェクトは前記識別手段にตอบสนองすることを特徴とするコンピュータ・システム。

(2) 前記コンピュータ・システムはグラフィカル・ユーザ・インタフェースを表示する表示装置に接続可能であり、また前記動作を表現するのに使用される前記ビューの少なくとも一つは、前記グラフィカル・ユーザ・インタフェースを介して表示可能なビジュアル・ビューであり、さらに前記ビューの少なくとも一つに対する前記動作オブジェクトに定義された前記属性に該ビジュアル・ビューを生成するのに必要なデータが含まれることを

特徴とする上記(1)に記載のコンピュータ・システム。

(3) 動作の集合ビューを定義するメニュー定義から前記動作の集合ビューを構築する構築手段をさらに備え、前記メニュー定義は前記集合ビューによって表現される動作オブジェクトおよび(または)他のメニュー定義を識別する項目を有し、さらに、前記構築手段は前記集合ビューを構築するために前記動作オブジェクトおよび(または)前記他のメニュー定義を参照するように配列されていることを特徴とする上記(1)または(2)に記載のコンピュータ・システム。

(4) 前記メニュー定義は、いつ集合ビューがユーザによる選択に対して表示またはイネーブルされなければならないかを定めるコンテキスト情報をさらに含むことを特徴とする上記(3)に記載のコンピュータ・システム。

(5) 前記動作オブジェクトに定義された前記識別子は、前記動作オブジェクトの固有識別子と、前記動作オブジェクトの選択に対して前記所望の操作を呼び出す呼び出し機構とを持つことを特徴とする上記(1)ないし(3)のいずれか一項に記載のコンピュータ・システム。

(6) 使用可能な動作のリストを提供するために、前記表示装置に表示されることが可能な動作パレット・ビューをさらに有することを特徴とする上記(2)ないし(5)のいずれか一項に記載のコンピュータ・システム。

(7) ユーザ・インタフェースを有し、かつ該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、さらに前記動作は前記ユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システムを操作する方法であって、(a)動作を表現するために、前記動作を表現するのに用いることが可能な前記一以上のビューの各々について、該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと、前記動作の選択によって呼び出される操作を識別する識別子とを提供し、さらに前記識別手段にตอบสนองして前記所望の操作を呼び出すコマンドを生成するために前記動作オブジェクトによって前記識別子が使用されるステップと、(b)ビューのユーザ選択がいつ行われたかを識別するステップと、(c)ビューのユーザ選択の識別にตอบสนองして、前記選択に対応する動作オブジェクトを伝えることによって、所望の操作を呼び出すステップと、を有することを特徴とするコンピュータ・システムの操作方法。

(8) 前記コンピュータ・システムはグラフィカル・ユーザ・インタフェースを表示する表示装置に接続可能であり、また前記動作を表現するのに使用される前記ビューの少なくとも一つは、前記グラフィカル・ユーザ・インタフェースを介して表示可能なビジュアル・ビューで

あり、さらに前記ビューの少なくとも一つに対する前記動作オブジェクトに定義された前記属性に該ビジュアル・ビューを生成するのに必要なデータが含まれることを特徴とする上記(7)に記載のコンピュータ・システムの操作方法。

(9) 動作の集合ビューを定義するメニュー定義から前記動作の集合ビューを構築するステップをさらに有し、前記メニュー定義は前記集合ビューによって表現される動作オブジェクトおよび(または)他のメニュー定義を識別する項目を有し、さらに、前記構築手段は前記集合ビューを構築するために前記動作オブジェクトおよび

(または)前記他のメニュー定義を参照するように配列されていることを特徴とする上記(7)に記載のコンピュータ・システムの操作方法。

(10) 前記メニュー定義は、いつ集合ビューがユーザによる選択に対して表示または使用可能とならなければならないかを定めるコンテキスト情報をさらに含むことを特徴とする上記(9)に記載のコンピュータ・システムの操作方法。

(11) 使用可能な動作のリストを提供するために、前記表示装置に表示されることが可能な動作パレット・ビューをさらに有することを特徴とする上記(7)ないし(10)のいずれか一項に記載のコンピュータ・システムの操作方法。

(12) アプリケーションを実行するためのプロセッサと、前記アプリケーションによって使用されるデータを格納するメモリと、ユーザ・インタフェースとを備え、該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、また前記動作は前記ユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システムに使用されるプログラム・プロダクトであって、動作を表現するために前記動作を表現するのに用いることが可能な前記一以上のビューの各々について該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと前記動作の選択によって呼び出される操作を識別する識別子とを提供する手段と、ビューのユーザ選択がいつ生じたかを識別し、前記選択の対応動作オブジェクトを伝える識別手段とを備え、さらに、前記識別子を使用して前記所望の操作を呼び出すコマンドを生成するために前記動作オブジェクトは前記識別手段に回答すること

を特徴とするプログラム・プロダクト。

(13) ユーザ・インタフェースを備え、かつ該ユーザ・インタフェースを介してユーザが所望の操作を呼び出すために動作を選択することが可能であり、また前記動作は前記ユーザ・インタフェースを介して一以上のビューによって選択可能であるコンピュータ・システム上で使用されることを目的としたアプリケーションを開発するためにコンピュータ・システム上で使用され、さらに前記アプリケーションを開発するために前記コンピュー

タ・システムにロードされた場合に以下の機能を提供するように設けられたプログラム・プロダクトであって、動作を表現するために前記動作を表現するのに用いることが可能な前記一以上のビューの各々について該ビューを与えるのに必要とされる属性を定義する動作オブジェクトと前記動作の選択によって呼び出される操作を識別する識別子とを生成する手段とを備え、さらに、前記識別子を使用して前記所望の操作を呼び出すコマンドを生成するために前記動作オブジェクトは前記識別手段に回答することを特徴とするプログラム・プロダクト。

#### 【図面の簡単な説明】

【図1】本発明の好ましい実施形態例にもとづくシステムを説明するためのブロック図である。

【図2】本発明の好ましい実施形態例にもとづくシステムを説明するためのブロック図である。

【図3】本発明の好ましい実施形態例にもとづいて、動作オブジェクトおよびメニュー定義を用いて集合ビューを生成するのに必要とされるオブジェクトおよび相互関係を説明するためのオブジェクトを示すブロック図である。

【図4】ユーザが集合ビュー内で選択を生成し、対応した操作が呼び出される際の制御の流れを説明するためのブロック図である。

【図5】動作を表すのに用いられる視覚的ビューの一例を説明するための平面図である。

【図6】動作を表すのに用いられる視覚的ビューの一例を説明するための平面図である。

【図7】動作を表すのに用いられる視覚的ビューの一例を説明するための平面図である。

【図8】動作を表すのに用いられる視覚的ビューの一例を説明するための平面図である。

【図9】動作を表すのに用いられる視覚的ビューの一例を説明するための平面図である。

【図10】動作を表すのに用いられる視覚的ビューの一例を説明するための平面図である。

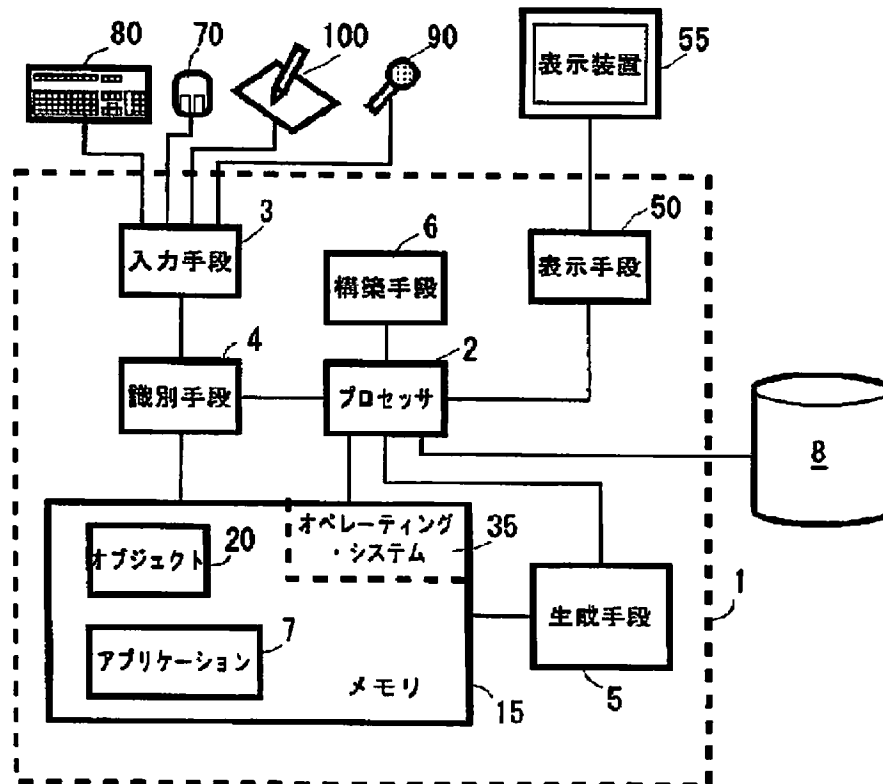
#### 【符号の説明】

2	プロセッサ (CPU)
3	入力手段
4	識別手段
5	生成手段
6	構築手段
7	アプリケーション
10	動作フレームワーク
15	メモリ
20	オブジェクト
30	ファイル・システム
35	オペレーティング・システム (OS)
、	
40	表示マネージメント
50	表示手段 (表示インタフェース)

25  
 55 表示装置  
 70 マウス入力装置  
 80 キーボード

26  
 90 スピーチ入力装置  
 100 ペン入力装置  
 110 イベント・マネージメント

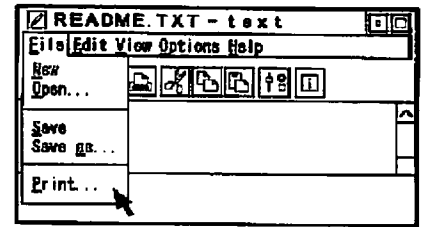
【図1】



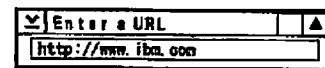
【図6】

【図7】

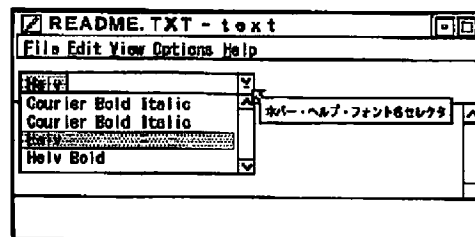
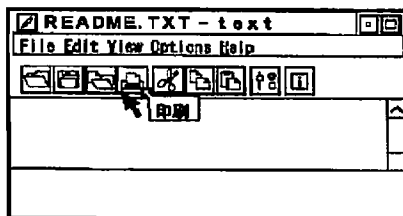
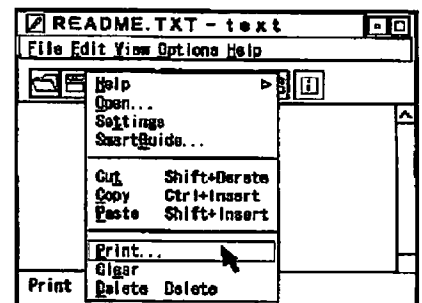
【図5】



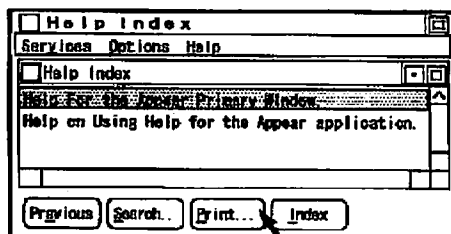
【図8】



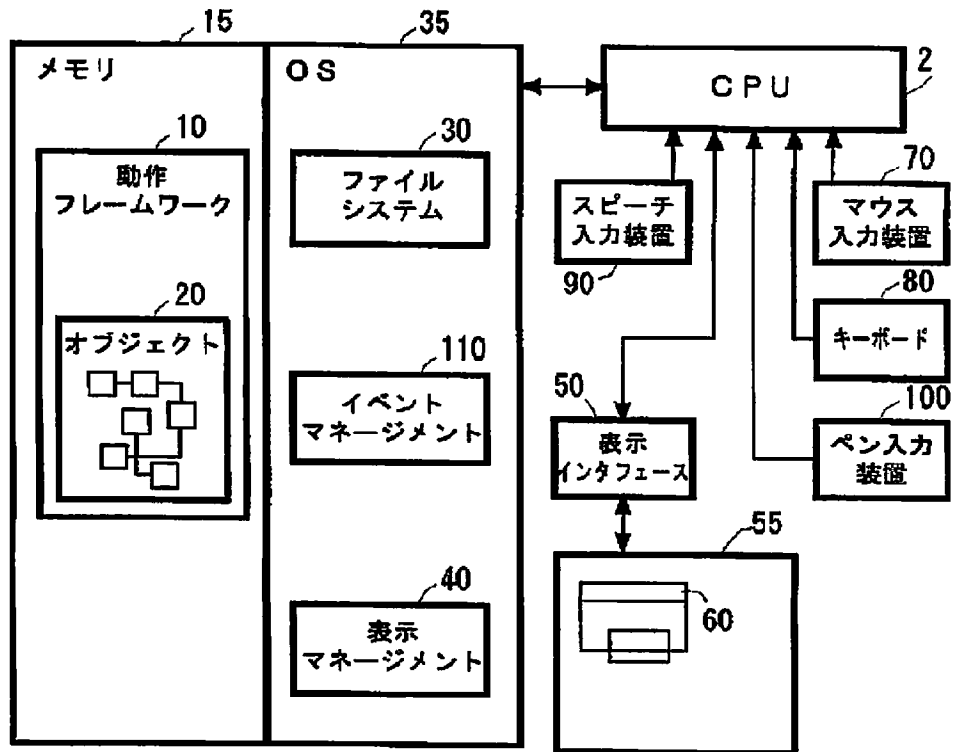
【図9】



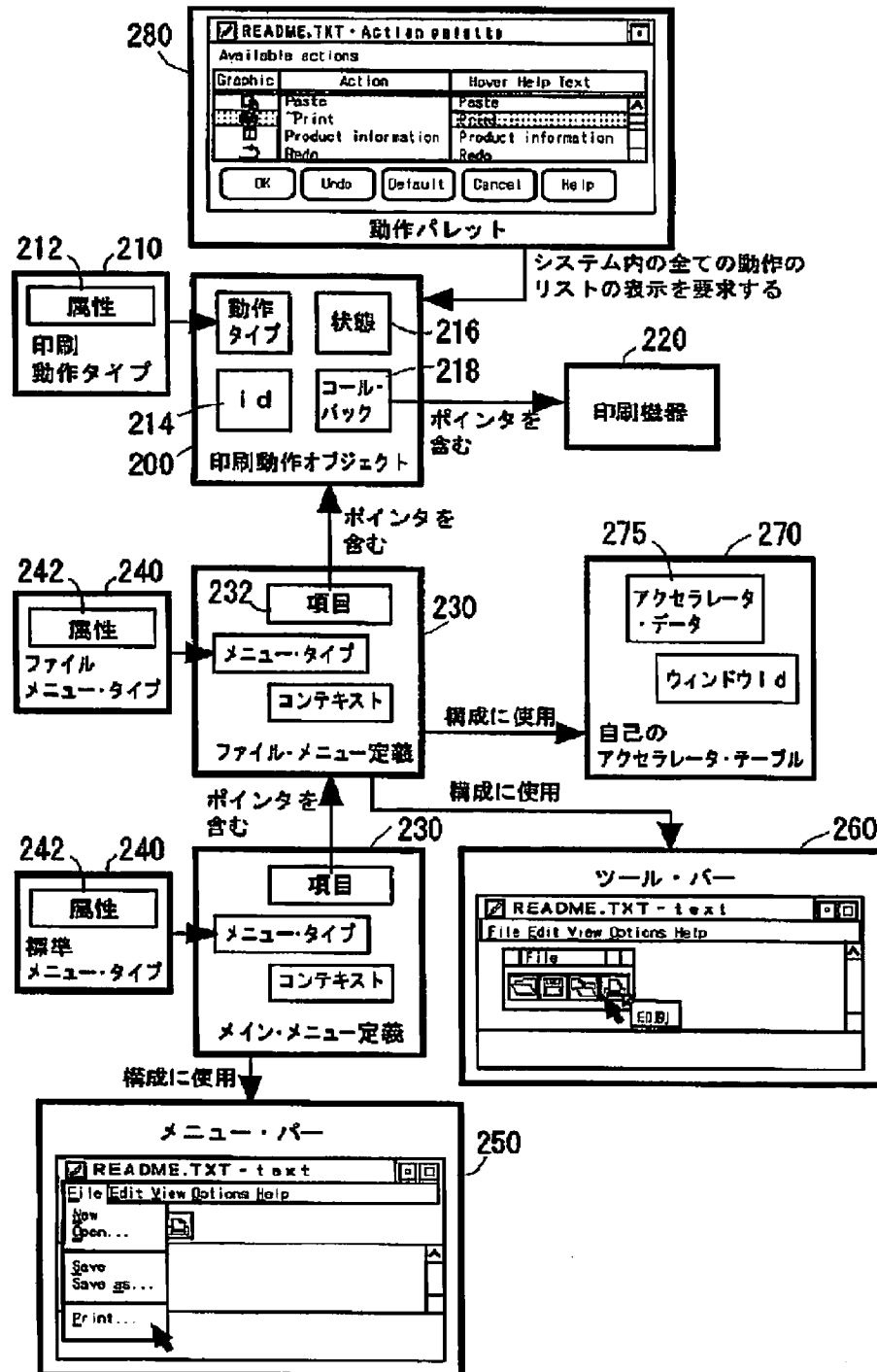
【図10】



【図2】

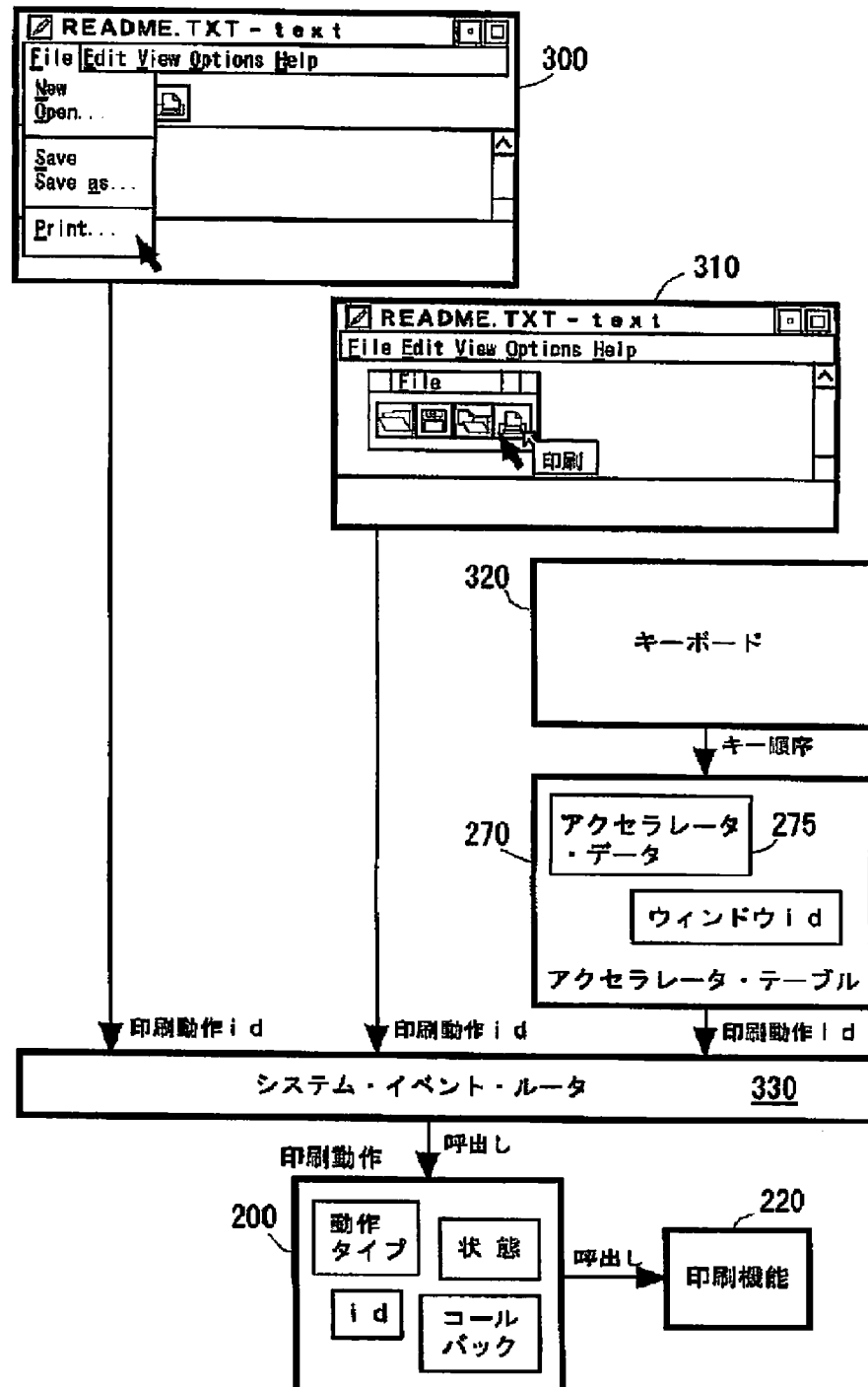


【図3】





【図4】



フロントページの続き

(72)発明者 レイ・トレイナー  
イギリス国 ウォリックシャー州 ウァー  
ウィックサムール ウェイ 25